lab2 report

于新雨 2022010841 计25

实现

重写 sys_get_time 和 sys_trace：
sys_get_time 问题是，传入的 ts 是用户态虚拟地址，在内核页表中没做映射，需要先手动映射为物理地址再做赋值

sys_trace 在进行 trace_read/trace_write 的时候，也需要先手动转换物理地址

mmap 和 munmap 实现：在 mmap 中，需要先检查一步该 range 是否被 mapped 过，如果被 mapped 过就需要返回 -1，同理，munmap 也需要先检查一步 unmap 的区域是否是被 mmap 出来的
然后 mmap 通过 TASK_MANAGER 拿到当前 task，通过 TaskControlBlock 的 memory_set 去插入新的 mmap 的虚拟地址范围，munmap 也是同样的实现，通过 current task 的 memory_set 来 unmap

问答作业

1. 请列举 SV39 页表页表项的组成，描述其中的标志位有何作用

```
63      54 53        28 27        19 18        10 9  8 7 6 5 4 3 2 1 0
+----------+------------+------------+------------+----+-+-+-+-+-+-+-+-+
| 保留位   |   PPN[2]   |   PPN[1]   |   PPN[0]   |保留|D|A|G|U|X|W|R|V|
+----------+------------+------------+------------+----+-+-+-+-+-+-+-+-+
```

标志位如下：
V： 页表项是否有效
R： 映射的页面是否可读 R：映射的页面是否可读
W：映射的页面是否可写
X：映射的页面是否可执行
U：用户态是否可以访问该页面
G：全局标志，表示该页是否对所有地址空间可见
A：访问标志，表示该页是否被访问过
D：脏标志，表示该页是否被写入过

2. 1. 请问哪些异常可能是缺页导致的？发生缺页时，描述相关重要寄存器的值 缺页可能导致以下异常：
   - 页面不存在（Page Fault）：访问的虚拟地址未被映射到物理地址
   - 权限异常：尝试以不允许的方式访问页面（如写入只读页面）

   发生缺页时，相关重要寄存器的值如下：

   - `stval`：保存导致异常的虚拟地址
   - `scause`：保存异常原因（如页面不存在或权限异常）
   - `satp`：保存当前页表基地址，用于定位页表项

- `sepc`：保存异常发生时的程序计数器，用于返回异常前的执行位置
- `stvec`: 保存 trap handler 地址

2. 缺页用 Lazy 策略处理有什么好处？ 便于推迟内存分配和页面加载的时机，仅在页面真正被访问时才进行处理。能够减少不必要的内存分配和 I/O 操作，从而提高性能和资源利用率。此外，Lazy 策略还可以更好地适应程序的实际运行需求，避免预分配可能浪费的内存

3. 处理 10G 连续的内存页面，对应的 SV39 页表大致占用多少内存 (估算数量级即可)？ 页面数为 10*2^18，第0级页表，每个页表可索引 512 个页面，需要 10*2^9 个页表，共 20MB 左右
第一级页表需要10个，共 40 KB 左右，第二级页表需要一个，4KB 所以总共是 20MB 的数量级

4. 如果 mmap 实现 Lazy 策略，请简单思考如何才能实现 Lazy 策略，缺页时又如何处理？描述合理即可，不需要考虑实现。 实现 Lazy 策略时，mmap 仅记录虚拟地址范围和相关标志位，不立即分配物理内存。缺页时，通过页表异常捕获未映射的虚拟地址，分配物理页面并更新页表项，同时将页面标记为有效。

5. 如果是用了 swap 策略，内存页面可能被换到磁盘上了，此时页面失效会表现为页表项 (PTE) 中的有效位 (V) 被清除，同时可以利用页表项中的保留位或其他字段记录页面在磁盘上的位置，以便在页面被访问时能够正确加载回内存。

3. 单页表意为，用户线程和对应的内核线程共用同一张页表，只不过内核对应的地址只允许在内核态访问

1. 在单页表情况下，如何更换页表 保存当前上下文，刷新 TLB，更新页表基址寄存器（satp），sfence.vma 然后恢复上下文
2. 如何控制用户态无法访问内核页面 通过看当前特权级，如果是 U mode ，则看访问页表表项是是否是 U 置位了，如果 U = 1 才可访问
3. 单页表优势 实现简洁，进内核态可以直接查当前页表，无需像实验中一样查用户态页表进行手动地址转换，性能更好
4. 单页表/双页表切换页表时机 双页表切换页表时机是用户态与内核态转换和进程切换时
单页表切换时机是进程切换和新进程创建时

honor code

1. 在完成本次实验的过程（含此前学习的过程）中，我曾分别与 以下各位 就（与本次实验相关的）以下方面做过交流，还在代码中对应的位置以注释形式记录了具体的交流对象及内容：

和郝子胥学长关于 TimeVal 的地址转换的问题理解、和 MemorySet 映射地址空间的具体接口使用上做过交流，并且记录了交流对象和内容

2. 此外，我也参考了 以下资料 ，还在代码中对应的位置以注释形式记录了具体的参考来源及内容：

-

3. 我独立完成了本次实验除以上方面之外的所有工作，包括代码与文档。 我清楚地知道，从以上方面获得的信息在一定程度上降低了实验难度，可能会影响起评分。

4. 我从未使用过他人的代码，不管是原封不动地复制，还是经过了某些等价转换。 我未曾也不会向他人（含此后各届同学）复制或公开我的实验代码，我有义务妥善保管好它们。 我提交至本实验的评测系统的代码，均无意于破坏或妨碍任何计算机系统的正常运转。 我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按“-100”分计。